

Chapter 16

GRAPH MINING APPLICATIONS TO SOCIAL NETWORK ANALYSIS

Lei Tang and Huan Liu

Computer Science & Engineering
Arizona State University

L.Tang@asu.edu, Huan.Liu@asu.edu

Abstract The prosperity of Web 2.0 and social media brings about many diverse social networks of unprecedented scales, which present new challenges for more effective graph-mining techniques. In this chapter, we present some graph patterns that are commonly observed in large-scale social networks. As most networks demonstrate strong community structures, one basic task in social network analysis is community detection which uncovers the group membership of actors in a network. We categorize and survey representative graph mining approaches and evaluation strategies for community detection. We then present and discuss some research issues for future exploration.

Keywords: Social Network Analysis, Graph Mining, Community Detection,

1. Introduction

Social Network Analysis (SNA) [61] is the study of relations between individuals including the analysis of social structures, social position, role analysis, and many others. Normally, the relationship between individuals, e.g., kinship, friends, neighbors, etc. are presented as a network. Traditional social science involves the circulation of questionnaires, asking respondents to detail their interaction with others. Then a network can be constructed based on the response, with nodes representing individuals and edges the interaction between them. This type of data collection confines traditional SNA to a limited scale, typically at most hundreds of actors in one study.

With the prosperity of Internet and Web 2.0, many social networking and social media sites are emerging, and people can easily connect to each other in the cyber space. This also facilitates SNA to a much larger scale — millions of actors or even more in a network; Examples include email communication networks [18], instant messenger networks [33], mobile call networks [39], friends networks [38]. Other forms of complex network, like coauthorship or citation networks [56], biological networks, metabolic pathways, genetic regulatory networks, food web and neural networks, are also examined and demonstrate similar patterns [44]. These large scale networks of various entities yield patterns that are normally not observed in small networks. In addition, they also pose computational challenges as well as new tasks and problems for the SNA.

Social network analysis involves a variety of tasks. To name a few, we list some that are among the most relevant to the data mining field:

- Centrality analysis aims to identify the “most important” actors in a social network. Centrality is a measure to calibrate the “importance” of an actor. This helps to understand the social influence and power in a network.
- Community detection. Actors in a social network form groups¹. This task identify these communities through the study of network structures and topology.
- Position/Role analysis identifies the role associated with different actors during network interaction. For instance, what is the role of “husband”? Who serves as the bridge between two groups?
- Network modeling attempts to simulate the real-world network via simple mechanisms such that the patterns presented in large-scale complex networks can be captured.
- Information diffusion studies how the information propagates in a network. Information diffusion also facilitates the understanding the cultural dynamics, and infection blocking.
- Network classification and outlier detection. Some actors are labeled with certain information. For instance, in a network with some terrorists identified, is it possible to infer other people who are likely to be terrorists by leveraging the social network information?
- Viral marketing and link prediction. The modeling of the information diffusion process, in conjunction with centrality analysis and communi-

¹In this chapter, community and group are used interchangeably.

ties, can help achieve more cost-effective viral marketing. That is, only a small set of users are selected for marketing. Hopefully, their adoption can influence other members in the network, so the benefit is maximized.

Normally, a social network is represented as a graph. How to mine the patterns in the graph for the above tasks becomes a hot topic thanks to the availability of enormous social network data. In this chapter, we attempt to present some recent trends of large social networks and discuss graph mining applications for social network analysis. In particular, we discuss graph mining applications to community detection, a basic task in SNA to extract meaningful social structures or positions, which also serves as basis for some other related SNA tasks. Representative approaches for community detection are summarized. Interesting emerging problems and challenges are also presented for future exploration.

For convenience, we define some notations used throughout this chapter. A network is normally represented as a graph $G(V, E)$, where V denotes the vertexes (equivalently nodes or actors) and E denotes edges (ties or connections). The connections are represented via adjacency matrix A , where $A_{ij} \neq 0$ denotes $(v_i, v_j) \in E$, while $A_{ij} = 0$ denotes $(v_i, v_j) \notin E$. The degree of node v_i is d_i . If the edges between nodes are directed, the in-degree and out-degree are denoted as d_i^- and d_i^+ respectively. Number of vertexes and edges of a network are $|V| = n$, and $|E| = m$, respectively. The shortest path between a pair of nodes v_i and v_j is called *geodesic*, and the geodesic distance between the two is denoted as $d(i, j)$. $G_s(V_s, E_s)$ represents a subgraph in G . The neighbors of a node v are denoted as $N(v)$. In a directed graph, the neighbors connecting to and from one node v are denoted as $N^-(v)$ and $N^+(v)$, respectively. Unless specified explicitly, we assume a network is unweighted and undirected.

2. Graph Patterns in Large-Scale Networks

Most large-scale networks share some common patterns that are not noticeable in small networks. Among all the patterns, the most well-known characteristics are: *scale-free distribution*, *small world effect*, and *strong community structure*.

2.1 Scale-Free Networks

Many statistics in real-world have a typical “scale”, a value around which the sample measurements are centered. For instance, the height of all the people in the United States, the speed of vehicles on a highway, etc. But the node degrees in real-world large scale social networks often follow a power law distribution (a.k.a. Zipfian distribution, Pareto distribution [41]). A random

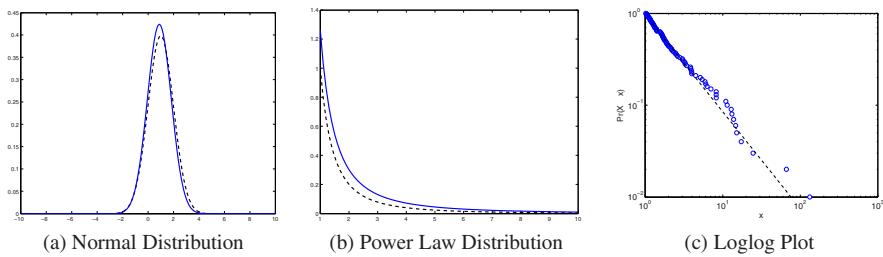


Figure 16.1. Different Distributions. A dashed curve shows the true distribution and a solid curve is the estimation based on 100 samples generated from the true distribution. (a) Normal distribution with $\mu = 1$, $\sigma = 1$; (b) Power law distribution with $x_{min} = 1$, $\alpha = 2.3$; (c) Loglog plot, generated via the toolkit in [17].

variable X follows a power law distribution if

$$p(x) = Cx^{-\alpha}, \quad x \geq x_{min}, \quad \alpha > 1 \quad (2.1)$$

here $\alpha > 1$ is to ensure a normalization constant C exists [41]. A power law distribution is also called *scale-free* distribution [8] as the shape of the distribution remains unchanged except for an overall multiplicative constant when the scale of units is increased by a factor. That is,

$$p(ax) = bp(x) \quad (2.2)$$

where a and b are constants. In other words, there is no characteristic scale with the random variable. The functional form is the same for all the scales. The network with a scale-free distribution for nodal degrees is also called *scale-free network*.

Figures 16.1a and 16.1b demonstrate a normal distribution and a power-law distribution respectively. While the normal distribution has a “center”, the power law distribution is highly skewed. For normal distribution, it is extremely rare for an event to occur that are several deviations away from the mean. On the contrary, power law distribution allows the tail to be much longer. That is, it is common that some nodes in a social network have extremely high degrees while the majority have few connections. The reason is that the decay of the tail for a power law distribution is polynomial. It is asymptotically slower than exponential as presented in the decay of normal distribution, resulting in a heavy-tail (or long-tail [6], fat-tail) phenomenon.

The curve of power law distribution becomes a straight line if we plot the degree distribution in a log-log scale, since

$$\log p(x) = -\alpha \log x + \log C$$

This is commonly used by practitioners to rigorously verify whether a distribution follows power law, though some researchers advise more careful statistical

examination to fit a power law distribution [17]. It can be verified the cumulative distribution function (cdf) can also be written in the following form:

$$F(X \geq x) \propto x^{-\alpha+1}$$

The samples of rare events (say, extremely high degrees in a network) are scarce, resulting in an unreliable estimation of the density. A more robust estimation is to approximate the cdf. One example of the loglog plot of cdf estimation is shown in Figure 16.1c.

Besides node degrees, some other network statistics are also observed to follow a power law pattern, for example, the largest eigenvalues of the adjacency matrix derived from a network [21], the size of connected components in a network [31], the information cascading size [36], and the densification of a growing network [34]. Scale-free distribution seems common rather than “by chance” for large-scale networks.

2.2 Small-World Effect

Travers and Milgram [58] conducted a famous experiment to examine the average path length for social networks of people in the United States. In the experiments, the subjects involved were asked to send a chain letter to his acquaintances starting from an individual in Omaha, Nebraska or Wichita, Kansas to the target individual in Boston, Massachusetts. Finally, 64 letters arrived and the average path length fell around 5.5 or 6, which later led to the so-called “six degrees of separation”. This result is also confirmed recently in a planetary-scale instant messaging network of more than 180 million people, in which the average path length of two messengers is 6.6 [33].

This small world effect is observed in many large scale networks. That is, two actors in a huge network are actually not too far away. To quantify the effect, different network measures are used:

- **Diameter:** a shortest path between two nodes is called a *geodesic*, and *diameter* is the length of the longest geodesic between any pair of nodes in the graph [61]. It might be the case that a network contains more than one connected component. Thus, no path exists between two nodes in different components. In this case, practitioners typically examine the geodesic between nodes of the same component. The diameter is the minimum number of hops required to reach all the connected nodes from any node.
- **Effective Eccentricity:** the minimum number of hops required to reach at least 90% of all connected pairs of nodes in the network [57]. This measure removes the effect of outliers that are connected through a long path.

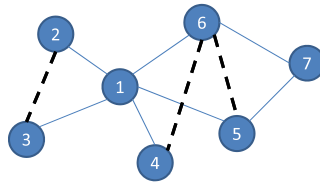


Figure 16.2. A toy example to compute clustering coefficient: $C_1 = 3/10$, $C_2 = C_3 = C_4 = 1$, $C_5 = 2/3$, $C_6 = 3/6$, $C_7 = 1$. The global clustering coefficient following Eqs. (2.5) and (2.6) are 0.7810 and 0.5217, respectively.

- Characteristic Path Length:** the median of the means of the shortest path lengths connecting each node to all other nodes (excluding unreachable ones) [12]. This measure focuses on the average distance between pairs rather than the maximum one as the diameter.

All the above measures involve the calculation of the shortest path between all pairs of connected nodes. Two simple approaches to compute the diameter are:

- Repeated matrix multiplication.** Let A denotes the adjacency matrix of a network, then the non-zero entries in A^k denote those pairs that are connected in k hops. The diameter corresponds to the minimum k so that all entries of A^k are non-zero. It is evident that this process leads to denser and denser matrix, which requires $O(n^2)$ space and $O(n^{2.88})$ time asymptotically for matrix multiplication.
- Breadth-first search** can be conducted starting from each node until all or a certain proportion (90% as for effective eccentricity) of the network nodes are reached. This costs $O(n + m)$ space but $O(nm)$ time.

Evidently, both approaches above become problematic when the network scales to millions of nodes. One natural solution is to sample the network, but it often leads to poor approximation. A randomized algorithm achieving better approximation is presented in [48].

2.3 Community Structures

Social networks demonstrate a strong community effect. That is, a group of people tend to interact with each other more than those outside the group. To measure the community effect, one related concept is *transitivity*. In a simple form, friends of a friend are likely to be friends as well. *Clustering coefficient* is proposed specifically to measure the transitivity, the probability of connections between one vertex's neighboring friends.

Definition 2.1 (Clustering Coefficient). Suppose a node v_i has d_i neighbors, and there are k_i edges among these neighbors, then the clustering coefficient

is

$$C_i = \begin{cases} \frac{k_i}{d_i \times (d_i - 1) / 2} & d_i > 1 \\ 0 & d_i = 0 \text{ or } 1 \end{cases} \quad (2.3)$$

The denominator is essentially the possible number of edges between the neighbors. Take the network in Figure 16.2 as an example. Node v_1 has 5 neighbors $v_2, v_3, v_4, v_5,$ and v_6 . Among these neighbors, there are 3 edges (dashed lines) $(v_2, v_3), (v_4, v_6)$ and (v_5, v_6) . Hence, the clustering coefficient of v_1 is $3/10$. Alternatively, clustering coefficient can also be equally defined as:

$$C_i = \frac{\text{number of triangles connected to node } v_i}{\text{number of connected triples centered on node } v_i} \quad (2.4)$$

where a triple is a tuple $(v_i, \{v_j, v_k\})$ such that $(v_i, v_j) \in E, (v_i, v_k) \in E,$ and the flanking nodes v_j and v_k are unordered. For instance, $(v_1, \{v_3, v_6\})$ and $(v_1, \{v_6, v_3\})$ in Figure 16.2 represent the same triple centered on v_1 and there are in total 10 such triples. Triangle denotes an *unordered set* of three vertexes such that each two is connected. The triangles connected to node v_1 are $\{v_1, v_2, v_3\}, \{v_1, v_4, v_6\}$ and $\{v_1, v_5, v_6\}$, so $C_1 = 3/10$.

To measure the community structure of a network, two commonly used global clustering coefficients are defined by extending the definition of Eqs. (2.3) and (2.4), respectively.

$$C = \sum_{i=1}^n C_i / n \quad (2.5)$$

$$\begin{aligned} C &= \frac{\sum_{i=1}^n \text{number of triangles connected to node } v_i}{\sum_{i=1}^n \text{number of connected triples centered on node } v_i} \\ &= \frac{3 \times \text{number of triangles in the network}}{\text{number of connected triples of nodes}} \end{aligned} \quad (2.6)$$

Eq. (2.5) yields high variance for nodes with less degrees. E.g., for nodes with degree 2, C_i is either 0 or 1. It is commonly used for numerical study [62] whereas Eq. (2.6) is used more for analytical study. In the toy example, the global clustering coefficients based the two formulas are 0.7810 and 0.5217 respectively.

The computation of global clustering coefficient relies on exact counting of triangles in the network which can be computationally expensive [5, 51, 30]. One efficient exact counting method without huge memory requirement is the simple node-iterator (or edge-iterator) algorithm, which essentially traverse all the nodes (edges) to compute the number of triangles connected to each node (edge). Some approximation algorithms are proposed, which require one single pass [13] or multiple passes [9] of the huge edge file. It can be verified that the number of triangles is proportional to the sum of the cube of eigenvalues of

the adjacency matrix [59]. Thus, using the few top eigenvalues to approximate the number is also viable.

While clustering coefficient and transitivity concentrate on microscopic view of community effect, communities of macroscopic view also demonstrate intriguing patterns. In real-world networks, a giant component tends to form with the remaining being singletons and minor communities [28]. Even within the giant component, tight but almost trivial communities (connecting to the rest of the network through one or two edges) at very small scales are often observed. Most social networks lack well-defined communities in a large scale [35]. The communities gradually “blend in” the rest of the network as their size expands.

2.4 Graph Generators

As large scale networks demonstrate similar patterns, one interesting question is: what is the innate mechanism of these networks? A variety of graph and network generators have been proposed such that these patterns can be reproduced following some simple rules. The classical model is the random graph model [20], in which the edges connecting nodes are generated probabilistically via flipping a biased coin. It yields beautiful mathematical properties but does not capture the common patterns discussed above. Recently, Watts and Strogatz proposed a model mixing the random graph model and a regular lattice structure, producing small diameter and high clustering effect [62]; a preferential attachment process is presented in [8] to explain the power law distribution exhibited in real-world networks. These two pieces of seminal work stir renewed enthusiasm researching on pursuing graph generators to capture some other network patterns. For instance, the availability of dynamic network data enables the possibility to study how a network evolves and how its fundamental network properties vary over time. It is observed that many growing networks are becoming denser with average degrees increasing. Meanwhile, the effective diameter shrinks with the growth of a network [34]. These properties cannot be explained by the aforementioned network models. Thus, a forest-fire model is proposed. While many models focus on global patterns present in networks, the microscopic property of networks is also calling for alternative explanations [32]. Please refer to surveys [40, 14] for more detailed discussion.

3. Community Detection

As mentioned above, social networks demonstrate strong community effect. The actors in a network tend to form groups of closely-knit connections. The groups are also called communities, clusters, cohesive subgroups or modules in different context. Roughly speaking, individuals interact more frequently

within a group than between groups. Detecting cohesive groups in a social network (also termed as *community detection*) remains a core problem in social network analysis. Finding out these groups also helps for other related tasks of social network analysis. Various definitions and approaches are exploited for community detection. Briefly, the criteria of groups fall into four categories: node-centric, group-centric, network-centric, and hierarchy-centric. Below, we elucidate some representative methods in each category.

3.1 Node-Centric Community Detection

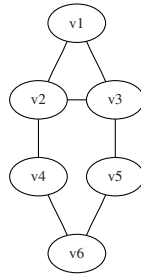
Community detection based on node-centric criteria requires *each node* in a group to satisfy certain properties like mutuality, reachability, or degrees.

Groups based on Complete Mutuality. An ideal cohesive group is a *clique*. It is a maximal complete subgraph of three or more nodes all of which are adjacent to each other. For a directed graph, [29] shows that with very high probability, there should exist a complete bipartite in a community. These complete bipartites work as a core for a community. The authors propose to extract an (i, j) -bipartite of which all the i nodes are connected to another j nodes in the graph.

Unfortunately, it is NP-hard to find out the maximum clique in a network. Even an approximate solution can be difficult to find. One brute-force approach to enumerate the cliques is to traverse of all the nodes in the network. For each node, check whether there is any clique of a specified size that contains the node. Then the clique is collected and the node is removed from future consideration. This works for small scale networks, but becomes impractical for large-scale networks. The main strategy to address this challenge is to effectively prune those nodes and edges that are unlikely to be contained in a maximal clique or a complete bipartite.

An algorithm to identify the maximal clique in large social networks is explored in [1]. Each time, a subset of the network is sampled. Based on this smaller set, a clique can be found in a greedy-search manner. The maximal clique found on the subset (say, it contains q nodes) serves as the lower bound for pruning. That is, the maximal clique should contain at least q members, so the nodes with degree less than q can be removed. This pruning process is repeated until the network is reduced to a reasonable size and the maximal clique can be identified.

A similar strategy can be applied to find complete bipartites. A subtle difference of the work in [29] is that it aims to find the complete bipartite of a fixed size, say an (i, j) -bipartite. Iterative pruning is applied to remove those nodes with their out-degree less than j and their in-degree less than i . After this initial pruning, an inclusion-exclusion pruning strategy is applied to either eliminate a node from concentration or discover an (i, j) -bipartite. The authors



- cliques: $\{v_1, v_2, v_3\}$
 2-cliques: $\{v_1, v_2, v_3, v_4, v_5\}, \{v_2, v_3, v_4, v_5, v_6\}$
 2-clans: $\{v_2, v_3, v_4, v_5, v_6\}$
 2-clubs: $\{v_1, v_2, v_3, v_4\}, \{v_1, v_2, v_3, v_5\}, \{v_2, v_3, v_4, v_5, v_6\}$

Figure 16.3. A toy example (reproduced from [61])

proposed to focus first on nodes that are of out-degree j (or of in-degree i). It is easy to check whether a node belongs to an (i, j) -bipartite by examining whether all its connected nodes have enough connections. So either one node is purged or an (i, j) -bipartite is identified.

Note that clique (or complete bipartite) is a *very* strict definition, and rarely can it be observed in a large size in real-world social networks. This structure is very unstable as the removal of any edge could break this definition. Practitioners typically use identified maximal cliques (or maximal complete bipartites) as cores or seeds for subsequent expansion for a community [47, 29]. Alternatively, other forms of substructures close to a clique are identified as communities as discussed next.

Groups based on Reachability. This type of community considers the reachability between actors. In the extreme case, two nodes can be considered as belonging to one community if there exists a path between the two nodes. Thus each component² is a community. This can be efficiently done in $O(n + m)$ time. However, in real-world networks, a giant component tends to form while many others are singletons and minor communities [28]. For those minorities, it is straightforward to identify them via connected components. More efforts are required to find communities in the giant component.

Conceptually, there should be a short path between any two nodes in a group. Several well studied structures in social science are:

- k -clique is a maximal subgraph in which the largest geodesic distance between any two nodes is no greater than k . That is,

$$d(i, j) \leq k \quad \forall v_i, v_j \in V_s$$

²Connected nodes form a component.

Note that the geodesic distance is defined on the original network. Thus, the geodesic is not necessarily included in the group structure. So a k -clique may have a diameter greater than k or even become disconnected.

- k -clan is a k -clique in which the geodesic distance $d(i, j)$ between all nodes in the subgraph is no greater than k for all paths within the subgraph. A k -clan must be a k -clique, but it is not so vice versa. For instance, $\{v_1, v_2, v_3, v_4, v_5\}$ in Figure 16.3 is a 2-clique, but not 2-clan as the geodesic distance of v_4 and v_5 is 2 in the original network, but 3 in the subgraph.
- k -club restricts the geodesic distance within the group to be no greater than k . It is a maximal substructure of diameter k .

All k -clans are k -cliques, and k -clubs are normally contained within k -cliques. These substructures are useful in the study of information diffusion and influence propagation.

Groups based on Nodal Degrees. This requires actors within a group to be adjacent to a relatively large number of group members. Two commonly studied substructures are:

- k -plex - It is a maximal subgraph containing n_s nodes, in which each node is adjacent to no fewer than $n_s - k$ nodes in the subgraph. In other words, each node may have no ties up to k group members. A k -plex becomes a clique when $k = 1$.
- k -core - It is a substructure that each node (v_i) connects to at least k members within the group, i.e.,

$$d_s(i) \geq k \quad \forall v_i \in V_s$$

The definitions of k -plex and k -core are actually complementary. A k -plex with group size equal to n_s , is also a $(n_s - k)$ -core. The structures above are normally robust to the removal of edges in the subgraph. Even if we miss one or two edges, the subgraph is still connected. Solving the k -plex and earlier k -clan problems requires involved combinatorial optimization [37]. As mentioned in the previous section, the nodal degree distribution in a social network follows power law, i.e., few nodes with many degrees and many others with few degrees. However, groups based on nodal degrees require all the nodes of a group to have at least a certain number of degrees, which is not very suitable for the analysis of large-scale networks where power law is a norm.

Groups based on Within-Outside Ties. This kind of group forces each node to have more connections to nodes that are within the group than to those outside the group.

- *LS sets*: A set of nodes V_s in a social network is an LS set iff *any of its proper subsets* has more ties to its complement within V_s than to those outside V_s . An important property which distinguishes LS sets from previous cliques, k -cliques and k -plexes, is that any two LS sets are either disjoint or one LS set contains the other [10]. This implies that a hierarchical series of LS sets exist in a network. However, due the strict constraint, large-size LS sets are rarely found in reality, leading to its limited usage for analysis. An alternative generalization is Lambda sets.
- *Lambda sets*: The group should be difficult to disconnect by the removal of edges in the subgraph. Let $\lambda(v_i, v_j)$ denote the number of edges that must be removed from the graph in order to disconnect any two nodes v_i and v_j . A set is called lambda set if

$$\lambda(v_i, v_j) > \lambda(v_k, v_\ell) \quad \forall v_i, v_j, v_k \in V_s, \forall v_\ell \in V \setminus V_s$$

It is a maximal subset of actors who have more edge-independent paths connecting them to each other than to outsiders. The minimum connectivity among the members of a lambda set is denoted as $\lambda(G_s)$.

There are more lambda sets in reality than LS sets, hence it is more practical to use lambda sets in network analysis. Akin to LS sets, lambda sets are also disjoint at an edge-connectivity level λ . To obtain a hierarchical structure of lambda sets, one can adopt a two-step algorithm:

- Compute the edge connectivity between any pair of nodes in the network via “maximum-flow, minimum-cut” algorithms.
- Starting from the highest edge connectivity, gradually join nodes such that $\lambda(v_i, v_j) \geq k$.

Since the lambda sets at each level (k) is disjoint, this generates a hierarchical structure of the nodes. Unfortunately, the first step is computationally prohibitive for large-scale networks as the minimum-cut computation involves each pair of nodes.

3.2 Group-Centric Community Detection

All of the above group definitions are node centric, i.e. each node in the group has to satisfy certain properties. Group-centric criteria, instead, consider the connections inside a group as whole. It is acceptable to have some nodes in the group to have low connectivity as long as the group overall satisfies certain requirements. One such example is *density-based groups*. A subgraph $G_s(V_s, E_s)$ is γ -dense (also called a quasi-clique [1]) if

$$\frac{E_s}{V_s(V_s - 1)/2} \geq \gamma \quad (3.1)$$

Clearly, the quasi-clique becomes a clique when $\gamma = 1$. Note that this density-based group typically does not guarantee the nodal degree or reachability for each node in the group. It allows the degree of different nodes to vary drastically, thus seems more suitable for large-scale networks.

In [1], the maximum γ -dense quasi-cliques are explored. A greedy algorithm is adopted to find a maximal quasi-clique. The quasi-clique is initialized with a vertex with the largest degree in the network, and then expanded with nodes that are likely to contribute to a large quasi-clique. This expansion continues until no nodes can be added to maintain the γ -density. Evidently, this greedy search for maximal quasi-clique is not optimal. So a subsequent local search procedure (GRASP) is applied to find a larger maximal quasi-clique in the local neighborhood. This procedure is able to detect a close-to-optimal maximal quasi-clique but requires the whole graph to be in main memory. To handle large-scale networks, the authors proposed to utilize the procedure above to find out the lower bound of degrees for pruning. In each iteration, a subset of edges are sampled from the network, and GRASP is applied to find a locally maximal quasi-clique. Suppose the quasi-clique is of size k , it is impossible to include in the maximal quasi-clique a node with degree less than $k\gamma$, all of whose neighbors also have their degree less than $k\gamma$. So the node and its incident edges can be pruned from the graph. This pruning process is repeated until GRASP can be applied directly to the remaining graph to find out the maximal quasi-clique.

For a directed graph like the Web, the work in [19] extends the complete-bipartite core [29] to γ -dense bipartite. (X, Y) is a γ -dense bipartite if

$$\forall x \in X, |N^+(x) \cap Y| \geq \gamma|Y| \quad (3.2)$$

$$\forall y \in Y, |N^-(y) \cap X| \geq \gamma'|X| \quad (3.3)$$

where γ and γ' are user provided constants. The authors derive a heuristic to efficiently prune the nodes. Due to the heuristic being used, not all satisfied communities can be enumerated. But it is able to identify some communities for a medium range of community size/density, while [29] favors to detect small communities.

3.3 Network-Centric Community Detection

Network-centric community detection has to consider the connections of the whole network. It aims to partition the actors into a number of disjoint sets. A group in this case is not defined independently. Typically, some quantitative criterion of the network partition is optimized.

Groups based on Vertex Similarity. Vertex similarity is defined in terms of how similar the actors interact with others. Actors behaving in the same

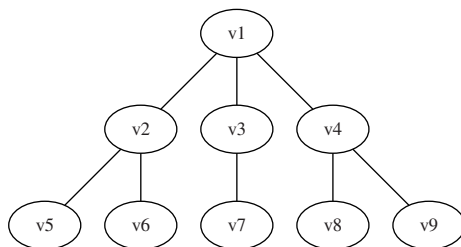


Figure 16.4. Equivalence for Social Position

role during interaction are in the same social position. The position analysis is to identify the social status and roles associated with different actors. For instance, what is the role of “wife”? What is the interaction pattern of “vice president” in a company organization? In position analysis, several concepts with decreasing strictness are studied to define two actors sharing the same social position [25]:

- **Structural Equivalence** Actors i and j are structurally equivalent, if for any actor k that $k \neq i, j$, $(i, k) \in E$ iff $(j, k) \in E$. In other words, actors i and j are connecting to exactly the same set of actors in the network. If the interaction is represented as a matrix, then rows (columns) i and j are the same except for the diagonal entries. For instance, in Figure 16.4, v_5 and v_6 are structurally equivalent. So are v_8 and v_9 .
- **Automorphic equivalence** Structural equivalence requires the connections of two actors to be exactly the same, yet it is too restrictive. Automorphic equivalence allows the connections to be isomorphic. Two actors u and v are automorphically equivalent iff all the actors of G can be relabeled to form an isomorphic graph. In the diagram, $\{v_2, v_4\}$, $\{v_5, v_6, v_8, v_9\}$ are automorphically equivalent, respectively.
- **Regular equivalence** Two nodes are regularly equivalent if they have the same profile of ties with other members that are also regularly equivalent. Specifically, u and v are regularly equivalent (denoted as $u \equiv v$) iff

$$(u, a) \in E \Rightarrow \exists b \in V, \text{ such that } (v, b) \in E \text{ and } a \equiv b \quad (3.4)$$

In the diagram, the regular equivalence results in three equivalence classes $\{v_1\}$, $\{v_2, v_3, v_4\}$, and $\{v_5, v_6, v_7, v_8, v_9\}$.

Structural equivalence is too restrictive for practical use, and no effective approach exists to scale automorphic equivalence or regular equivalence to more than thousands of actors. In addition, in large networks (say, online friends networks), the connection is very noisy. Meaningful equivalence of large scale is

difficult to detect. So some simplified similarity measures ignoring the social roles are used in practice, including cosine similarity [27], Jaccard similarity [23], etc. They consider the connections as features for actors, and rely on the fact that actors sharing similar connections tend to reside within the same community. Once the similarity measure is determined, classical k-means or hierarchical clustering algorithm can be applied.

It can be time consuming to compute the similarity between each pair of actors. Thus, Gibson et al. [23] present an efficient two-level shingling algorithm for fast computation of web communities. Generally speaking, the shingling algorithm maps each vector (the connection of actors) into a constant number of “shingles”. If two actors are similar, they share many shingles; otherwise, they share few. After initial shingling, each shingle is associated with a group of actors. In a similar vein, the shingling algorithm can be applied to the first-level shingles as well. So similar shingles end up sharing the same meta-shingles. Then all the actors relating to one meta-shingle form one community. This two-level shingling can be efficiently computed even for large-scale networks. Its time complexity is approximately linear to the number of edges. By contrast, normal similarity-based methods have to compute the similarity for each pair of nodes, totaling $O(n^2)$ time at least.

Groups based on Minimum-Cut. A community is defined as a vertex subset $C \subset V$, such that $\forall v \in C$, v has at least as many edges connecting to vertices in C as it does to vertices in $V \setminus C$ [22]. Flake et al. show that the community can be found via s - t minimum cut given a source node s in the community and a sink node t outside the community as long as both ends satisfy certain degree requirement. Some variants of minimum cut like normalized cut and ratio cut can be applied to SNA as well. Suppose we have a partition of k communities $\pi = (V_1, V_2, \dots, V_k)$, it follows that

$$\text{Ratio Cut}(\pi) = \sum_{i=1}^k \frac{\text{cut}(V_i, \bar{V}_i)}{|V_i|} \quad (3.5)$$

$$\text{Normalized Cut}(\pi) = \sum_{i=1}^k \frac{\text{cut}(V_i, \bar{V}_i)}{\text{vol}(V_i)} \quad (3.6)$$

where $\text{vol}(V_i) = \sum_{v_j \in V_i} d_j$. Both objectives attempt to minimize the number of edges between communities, yet avoid the bias of trivial-size communities like singletons. Interestingly, both formulas can be recast as an optimization problem of the following type:

$$\min_{S \in \{0,1\}^{n \times k}} \text{Tr}(S^T L S) \quad (3.7)$$

where L is the graph Laplacian (normalized Laplacian) for ratio cut (normalized cut), and $S \in \{0, 1\}^{n \times k}$ is a community indicator matrix defined below:

$$S_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ belongs to community } j \\ 0 & \text{otherwise} \end{cases}$$

Due to the discreteness property of S , this problem is still NP-hard. A standard way is to adopt a spectral relaxation to allow S to be continuous leading to the following trace minimization problem:

$$\min_{S \in \mathbb{R}^{n \times k}} \text{Tr}(S^T L S) \quad \text{s.t. } S^T S = I \quad (3.8)$$

It follows that S corresponds to the eigenvectors of k smallest eigenvalues (except 0) of Laplacian L . Note that a graph Laplacian always has an eigenvector $\mathbf{1}$ corresponding to the eigenvalue 0. This vector indicates all nodes belong to the same community, which is useless for community partition, thus is removed from consideration. The obtained S is essentially an approximation to the community structure. In order to obtain a disjoint partition, some local search strategy needs to be applied. An effective and widely used strategy is to apply k-means on the matrix S to find the partitions of actors.

The main computational cost with the above spectral clustering is that an eigenvector problem has to be solved. Since the Laplacian matrix is usually sparse, the eigenvectors corresponding to the smallest eigenvalues can be computed in an efficient way. However, the computational cost is still $O(n^2)$, which can be prohibitive for mega-scale networks.

Groups based on Block Model Approximation. Block modeling assumes the interaction between two vertices depends only on the communities they belong to. The actors within the same community are *stochastically equivalent* in the sense that the probabilities of the interaction with all other actors are the same for actors in the same community [46, 4]. Based on this block model, one can apply classical Bayesian inference methods like EM or Gibbs sampling to perform maximum likelihood estimation for the probability of interaction as well as the community membership of each actor.

In a different fashion, one can also use matrix approximation for block models. That is, the actors in the interaction matrix can be reordered in a form such that those actors sharing the same community form a dense interaction block. Based on the stochastic assumption, it follows that the community can be identified based on interaction matrix A via the following optimization [63]:

$$\min_{S, \Sigma} \ell(A; S^T \Sigma S) \quad (3.9)$$

Ideally, S should be an cluster indicator matrix with entry values being 0 or 1, Σ captures the strength of between-community interaction, and ℓ is the loss

function. To solve the problem, spectral relaxation of S can be adopted. If S is relaxed to be continuous, it is then similar to spectral clustering. If S is constrained to be non-negative, then it shares the same spirit as stochastic block models. This matrix approximation often resorts to numerical optimization techniques like alternating optimization or gradient methods rather than Bayesian inference.

Groups based on Modularity. Different from other criteria, modularity is a measure which considers the degree distribution while calibrating the community structure. Consider dividing the interaction matrix A of n vertices and m edges into k non-overlapping communities. Let s_i denote the community membership of vertex v_i , d_i represents the degree of vertex i . Modularity is like a statistical test that the null model is a uniform random graph model, in which one actor connects to others with uniform probability. For two nodes with degree d_i and d_j respectively, the expected number of edges between the two in a uniform random graph model is $d_i d_j / 2m$. Modularity measures how far the interaction is deviated from a uniform random graph. It is defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{d_i d_j}{2m} \right] \delta(s_i, s_j) \quad (3.10)$$

where $\delta(s_i, s_j) = 1$ if $s_i = s_j$. A larger modularity indicates denser within-group interaction. Note that Q could be negative if the vertices are split into bad clusters. $Q > 0$ indicates the clustering captures some degree of community structure.

In general, one aims to find a community structure such that Q is maximized. While maximizing the modularity over hard clustering is proved to be NP hard [11], a spectral relaxation of the problem can be solved efficiently [42]. Let $\mathbf{d} \in \mathbb{Z}_+^n$ be the degree vector of all nodes where \mathbb{Z}_+^n is the set of positive numbers of n dimensionality, $S \in \{0, 1\}^{n \times k}$ be a community indicator matrix, and the modularity matrix defined as

$$B = A - \frac{\mathbf{d}\mathbf{d}^T}{2m} \quad (3.11)$$

The modularity can be reformulated as

$$Q = \frac{1}{2m} \text{Tr}(S^T B S) \quad (3.12)$$

Relaxing S to be continuous, it can be shown that the optimal S is the top- k eigenvectors of the modularity matrix B [42].

Groups based on Latent Space Model. Latent space model [26, 50, 24] maps the actors into a latent space such that those with dense connections are

likely to occupy the latent positions that are not too far away. They assume the interaction between actors depends on the positions of individuals in the latent space. A maximum likelihood estimation can be utilized to estimate the position.

3.4 Hierarchy-Centric Community Detection

Another line of community detection is to build a hierarchical structure of communities based on network topology. This facilitates the examination of communities at different granularity. There are mainly three types of hierarchical clustering: divisive, agglomerative, and structure search.

Divisive hierarchical clustering. Divisive clustering first partitions the actors into several disjoint sets. Then each set is further divided into smaller ones until the set contains only a small number of actors (say, only 1). The key here is how to split the network into several parts. Some partition methods presented in previous section can be applied recursively to divide a community into smaller sets. One particular divisive clustering proposed for graphs is based on edge betweenness [45]. It progressively removes edges that are likely to be bridges between communities. If two communities are joined by only a few cross-group edges, then all paths through the network from nodes in one community to the other community have to pass along one of these edges. Edge betweenness is a measure to count how many shortest paths between pair of nodes pass along the edge, and this number is expected to be large for those between-group edges. Hence, progressively removing those edges with high betweenness can gradually disconnects the communities, which leads naturally to a hierarchical community structure.

Agglomerative hierarchical clustering. Agglomerative clustering begins with each node as a separate community and merges them successively into larger communities. Modularity is used as a criterion [15] to perform hierarchical clustering. Basically, a community pair should be merged if doing so results in the largest increase of overall modularity, and the merge continues until no merge can be found to improve the modularity. It is noticed that this algorithm incurs many imbalanced merges (a large community merges with a tiny community), resulting in high computational cost [60]. Hence, the merge criterion is modified accordingly to take into consideration the size of communities. In the new scheme, communities of comparable sizes are joined first, leading to a more balanced hierarchical structure of communities and to improved efficiency.

Structure Search. Structure search starts from a hierarchy and then searches for hierarchies that are more likely to generate the network. This idea

first appears in [55] to maintain a topic taxonomy for group profiling, and then a similar idea is applied for hierarchical construction of communities in social networks. [16] defines a random graph model for hierarchies such that two actors are connected based on the interaction probability of their least common ancestor node in the hierarchy. The authors generate a sequence of hierarchies via local changes of the network and accept it proportional to the likelihood. The final hierarchy is the consensus of a set of comparable hierarchies. The bottleneck with structure search approach is its huge search space. A challenge is how to scale it to large networks.

4. Community Structure Evaluation

In the previous section, we describe some representative approaches for community detection. Part of the reason that there are so many assorted definitions and methods, is that there is no clear ground truth information about the community structure in a real world network. Therefore, different community detection methods are developed from various applications of specific needs. In this section, we depict strategies commonly adopted to evaluate identified communities in order to facilitate the comparison of different community detection methods.

Depending on network information, different strategies can be taken for comparison:

- Groups with self-consistent definitions. Some groups like cliques, k-cliques, k-clans, k-plexes and k-cores can be examined immediately once a community is identified. If the goal of community detection is to enumerate all the desirable substructures of this sort, the total number of retrieved communities can be compared for evaluation.
- Networks with ground truth. That is, the community membership for each actor is known. This is an ideal case. This scenario hardly presents itself in real-world large-scale networks. It usually occurs for evaluation on synthetic networks (generated based on predefined community structures) [56] or a tiny network [42]. To compare the ground truth with identified community structures, visualization can be intuitive and straightforward [42]. If the number of communities is small (say 2 or 3 communities), it is easy to determine a one-to-one mapping between the identified communities and the ground truth. So conventional classification measures such as error-rate, F1-measure can be used. However, when there are a plurality of communities, it may not be clear what a correct mapping is. Instead, normalized mutual information (NMI) [52]

can be adopted to measure the difference of two partitions:

$$NMI(\pi^a, \pi^b) = \frac{\sum_{h=1}^{k^{(a)}} \sum_{\ell=1}^{k^{(b)}} n_{h,\ell} \log \left(\frac{n \cdot n_{h,\ell}}{n_h^{(a)} \cdot n_\ell^{(b)}} \right)}{\sqrt{\left(\sum_{h=1}^{k^{(a)}} n_h^{(a)} \log \frac{n_h^{(a)}}{n} \right) \left(\sum_{\ell=1}^{k^{(b)}} n_\ell^{(b)} \log \frac{n_\ell^{(b)}}{n} \right)}} \quad (4.1)$$

where π^a, π^b denotes two different partitions of communities. $n_{h,\ell}, n_h^a, n_\ell^b$ are, respectively, the number of actors simultaneously belonging to the h -th community of π^a and ℓ -th community of π^b , the number of actors in the h -th community of partition π^a , and the number of actors in the ℓ -th community of partition π^b . NMI is a measure between 0 and 1 and equals to 1 when π^a and π^b are the same.

- Networks with semantics. Some networks come with semantic or attribute information of the nodes and connections. In this case, the identified communities can be verified by human subjects to check whether it is consistent with the semantics. For instance, whether the community identified in the Web is coherent to a shared topic [22, 15], whether the clustering of coauthorship network captures the research interests of individuals. This evaluation approach is applicable when the community is reasonably small. Otherwise, selecting the top-ranking actors as representatives of a community is commonly used. This approach is qualitative and hardly can it be applied to all communities in a large network, but it is quite helpful for understanding and interpretation of community patterns.
- Networks without ground truth or semantic information. This is the most common situation, yet it requires objective evaluation most. Normally, one resorts to some quantitative measures for evaluation. One common measure being used is modularity [43]. Once we have a partition, we can compute its modularity. The method with higher modularity wins. Another comparable approach is to use the identified community as a base for link prediction, i.e., two actors are connected if they belong to the same community. Then, the predicted network is compared with the true network, and the deviation is used to calibrate the community structure. Since social network demonstrates strong community effect, a better community structure should predict the connections between actors more accurately. This is essentially checking how far the true network deviates from a block model based on the identified communities.

5. Research Issues

We have now described some graph mining techniques for community detection, a basic task in social network analysis. It is evident that community detection, though it has been studied for many years, is still in pressing need for effective graph mining techniques for large-scale complex networks. We present some key problems for further research:

- **Scalability.** One major bottleneck with community detection is scalability. Most existing approaches require a combinatorial optimization formulation for graph mining or eigenvalue problem of the network. Some alternative techniques are being developed to overcome the barrier, including local clustering [49] and multi-level methods [2]. How to find out meaningful communities efficiently and develop scalable methods for mega-scale networks remains a big challenge.
- **Community evolution.** Most networks tend to evolve over time. How to effectively capture the community evolution in dynamic social networks [56]? Can we find the members which act like the backbone of communities? How does this relate to the influence of an actor? What are the determining factors that result in community evolution [7]? How to profile the characteristics of evolving communities[55]?
- **Usage of communities.** How to utilize these communities for further social network analysis needs more exploration, especially for those emerging tasks in social media like classification [53], ranking, finding influential actors [3], viral marketing, link prediction, etc. Community structures of a social network can be exploited to accomplish these tasks.
- **Utility of patterns.** As we have introduced, large-scale social networks demonstrate some distinct patterns that are not usually observable in small networks. However, most existing community detection methods do not take advantage of the patterns in their detection process. How to utilize these patterns with various community detection methods remains unclear. More research should be encouraged in this direction.
- **Heterogeneous networks.** In reality, multiple relationships can exist between individuals. Two persons can be friends and colleagues at the same time. In online social media, people interact with each other in a variety of forms resulting in a multi-relational (multi-dimensional) network [54]. Some systems also involve multiple types of entities to interact with each other, leading to multi-mode networks [56]. Analysis of these heterogeneous networks involving heterogeneous actors or relations demands further investigation.

The prosperity of social media and emergence of large-scale complex networks poses many challenges and opportunities to graph mining and social network analysis. The development of graph mining techniques can facilitate the analysis of networks in a much larger scale, and help understand human social behaviors. Meanwhile, the common patterns and emerging tasks in social network analysis continually surprise us and stimulate advanced graph mining techniques. In this chapter, we point out the converging trend of the two fields and expect its healthy acceleration in the near future.

References

- [1] J. Abello, M. G. C. Resende, and S. Sudarsky. Massive quasi-clique detection. In *LATIN*, pages 598–612, 2002.
- [2] A. Abou-Rjeili and G. Karypis. Multilevel algorithms for partitioning power-law graphs. pages 10 pp.–, April 2006.
- [3] N. Agarwal, H. Liu, L. Tang, and P. S. Yu. Identifying the influential bloggers in a community. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 207–218, New York, NY, USA, 2008. ACM.
- [4] E. Airodi, D. Blei, S. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.
- [5] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.
- [6] C. Anderson. *The Long Tail: why the future of business is selling less of more*. 2006.
- [7] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, New York, NY, USA, 2006. ACM.
- [8] A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.
- [9] L. Becchetti, P. Boldi, C. Castillo, and A. Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–24, New York, NY, USA, 2008. ACM.
- [10] S. P. Borgatti, M. G. Everett, and P. R. Shirey. Ls sets, lambda sets and other cohesive subsets. *Social Networks*, 12:337–357, 1990.

- [11] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. Maximizing modularity is hard. *Arxiv preprint physics/0608255*, 2006.
- [12] T. Bu and D. Towsley. On distinguishing between internet power law topology generators. In *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 638–647 vol.2, 2002.
- [13] L. S. Buriol, G. Frahling, S. Leonardi, A. Marchetti-Spaccamela, and C. Sohler. Counting triangles in data streams. In *PODS '06: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 253–262, New York, NY, USA, 2006. ACM.
- [14] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2, 2006.
- [15] A. Clauset, M. Mewman, and C. Moore. Finding community structure in very large networks. *Arxiv preprint cond-mat/0408187*, 2004.
- [16] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.
- [17] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *arXiv*, 706, 2007.
- [18] J. Diesner, T. L. Frantz, and K. M. Carley. Communication networks from the enron email corpus "it's always about the people. enron is no different". *Comput. Math. Organ. Theory*, 11(3):201–228, 2005.
- [19] Y. Dourisboure, F. Geraci, and M. Pellegrini. Extraction and classification of dense communities in the web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 461–470, New York, NY, USA, 2007. ACM.
- [20] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5:17–61, 1960.
- [21] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 251–262, New York, NY, USA, 1999. ACM.
- [22] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160, New York, NY, USA, 2000. ACM.
- [23] D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *VLDB '05: Proceedings of the 31st inter-*

- national conference on Very large data bases*, pages 721–732. VLDB Endowment, 2005.
- [24] M. S. Handcock, A. E. Raftery, and J. M. Tantrum. Model-based clustering for social networks. *Journal Of The Royal Statistical Society Series A*, 127(2):301–354, 2007.
- [25] R. Hanneman and M. Riddle. *Introduction to Social Network Methods*. <http://faculty.ucr.edu/hanneman/>, 2005.
- [26] P. D. Hoff and M. S. H. Adrian E. Raftery. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.
- [27] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Natural communities in large linked networks. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 541–546, New York, NY, USA, 2003. ACM.
- [28] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 611–617, New York, NY, USA, 2006. ACM.
- [29] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. *Comput. Netw.*, 31(11-16):1481–1493, 1999.
- [30] M. Latapy. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theor. Comput. Sci.*, 407(1-3):458–473, 2008.
- [31] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 228–237, New York, NY, USA, 2006. ACM.
- [32] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470, New York, NY, USA, 2008. ACM.
- [33] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 915–924, New York, NY, USA, 2008. ACM.
- [34] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1):2, 2007.
- [35] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information net-

- works. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 695–704, New York, NY, USA, 2008. ACM.
- [36] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SIAM International Conference on Data Mining (SDM 2007)*, 2007.
- [37] B. McClosky and I. V. Hicks. Detecting cohesive groups. <http://www.caam.rice.edu/~ivhicks/CokplexAlgorithmPaper.pdf>, 2009.
- [38] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42, New York, NY, USA, 2007. ACM.
- [39] A. A. Nanavati, S. Gurusurthy, G. Das, D. Chakraborty, K. Dasgupta, S. Mukherjee, and A. Joshi. On the structural properties of massive telecom call graphs: findings and implications. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 435–444, New York, NY, USA, 2006. ACM.
- [40] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.
- [41] M. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary physics*, 46(5):323–352, 2005.
- [42] M. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3), 2006.
- [43] M. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.
- [44] M. Newman, A.-L. Barabasi, and D. J. Watts, editors. *The Structure and Dynamics of Networks*. 2006.
- [45] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.
- [46] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.
- [47] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.
- [48] C. R. Palmer, P. B. Gibbons, and C. Faloutsos. ANF: a fast and scalable tool for data mining in massive graphs. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 81–90, New York, NY, USA, 2002. ACM.

- [49] S. Papadopoulos, A. Skusa, A. Vakali, Y. Kompatsiaris, and N. Wagner. Bridge bounding: A local approach for efficient community discovery in complex networks. Feb 2009.
- [50] P. Sarkar and A. W. Moore. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, 7(2):31–40, 2005.
- [51] T. Schank and D. Wagner. Finding, counting and listing all triangles in large graphs, an experimental study. In *Workshop on Experimental and Efficient Algorithms*, 2005.
- [52] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.
- [53] L. Tang and H. Liu. Relational learning via latent social dimensions. In *KDD '09: Proceeding of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.
- [54] L. Tang and H. Liu. Uncovering cross-dimension group structures in multi-dimensional networks. In *SDM workshop on Analysis of Dynamic Networks*, 2009.
- [55] L. Tang, H. Liu, J. Zhang, N. Agarwal, and J. J. Salerno. Topic taxonomy adaptation for group profiling. *ACM Trans. Knowl. Discov. Data*, 1(4):1–28, 2008.
- [56] L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 677–685, New York, NY, USA, 2008. ACM.
- [57] S. Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the internet topology. In *Global Telecommunications Conference*, volume 3, pages 1667–1671, 2001.
- [58] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.
- [59] C. E. Tsourakakis. Fast counting of triangles in large real networks without counting: Algorithms and laws. *IEEE International Conference on Data Mining*, 0:608–617, 2008.
- [60] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks: [extended abstract]. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1275–1276, New York, NY, USA, 2007. ACM.
- [61] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
- [62] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

- [63] K. Yu, S. Yu, and V. Tresp. Soft clustering on graphs. In *NIPS*, 2005.